Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Midterm Exam Project

Topic:	**Module 2.0: Feature Extraction and Object Detection**	**Week No.**	10
Course Code:	**CSST106**	**Term:**	1st Semester
Course Title:	**Perception and Computer Vision**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date	**October 25, 2024**	**Points**	

Mid-term Project: Implementing Object Detection on a Dataset

For this project, students will be grouped into pairs. Here are the grouping guidelines:

1. **Group Size:**
   - Each group can have a **maximum of 2 members**.
   - **Single-member groups** (1 student) are allowed if needed.

2. **Forming Groups:**
   - Students can select their own partners if they prefer.
   - If a student wishes to work alone, they must inform the instructor in advance.

3. **Submission:**
   - Each group will submit **one project** under both members' names (or just one if it's a solo project).
   - Ensure the report, code, and video documentation include the names and section of all group members.

4. **Responsibilities:**
   - Both members of a group are expected to contribute equally to the project.
   - Single-member groups should manage their workload accordingly.

These guidelines ensure flexibility in group formation while maintaining a manageable project size.

Project Outline:

1. **Selection of Dataset and Algorithm:**

   - Each student will choose a dataset suitable for object detection tasks. The dataset can be from publicly available sources (e.g., COCO, PASCAL VOC) or one they create.

   - Select an object detection algorithm to apply to the chosen dataset. Possible algorithms include:
     - **HOG-SVM (Histogram of Oriented Gradients with Support Vector Machine):** Traditional method for object detection.
     - **YOLO (You Only Look Once):** A real-time deep learning-based approach.
     - **SSD (Single Shot MultiBox Detector):** A deep learning method balancing speed and accuracy.

2. **Implementation:**

   - **Data Preparation:** Preprocess the dataset by resizing images, normalizing pixel values, and, if necessary, labeling bounding boxes for objects.

   - **Model Building:** Implement the selected object detection algorithm using appropriate libraries (e.g., OpenCV for HOG-SVM, TensorFlow/Keras for YOLO or SSD).

   - **Training the Model:** Use the training data to train the object detection model. For deep learning methods, fine-tune hyperparameters (e.g., learning rate, batch size, epochs) to optimize model performance.

   - **Testing:** Evaluate the model on a test set to assess its detection capabilities. Ensure to capture edge cases where the model may struggle.

3. **Evaluation:**

   - **Performance Metrics:** Assess the model's performance using various metrics, including:

     - **Accuracy:** Overall success rate of object detection.

     - **Precision:** The proportion of true positive detections out of all positive predictions.

     - **Recall:** The proportion of true positive detections out of all actual positives in the dataset.

     - **Speed:** Measure the time taken for the model to detect objects in an image or video frame.

   - **Comparison:** Compare the results of the chosen model against other potential algorithms (e.g., how HOG-SVM compares to YOLO or SSD in terms of speed and accuracy).

4. **Submission Instructions:**

- **Repository Setup:**
  - Create a folder named Midterm-Project within your GitHub repository (**CSST106-Perception-and-Computer-Vision**).

- **File Organization:**
  - **code/**: Include all Python scripts or Jupyter notebooks.
  - **images/**: Store processed images showing detection results.
  - **documentation/**: Add the report (report.md or report.pdf).
  - **video/**: Include the video file documenting the project (video.mp4).

- **Filename Format:**
  - Use the format [**SECTION-LASTNAME-MP**] for files, e.g., *4D-Garcia-MP.py*, *4D-Garcia-MP-results.jpg*, *4D-Garcia-MP-video.mp4*.

- **Deadline:** Submit the project by the specified due date to avoid penalties for late submissions.

Guidelines for Video Documentation: Project from Start to Finish

Your video documentation is a crucial part of the project, as it showcases the entire process of creating the object detection model. Here's a breakdown of what the video should include, along with tips for a clear and comprehensive presentation:

Video Length: 10-15 minutes

1. Introduction (1-2 minutes)

- **Introduce the Project:** Briefly introduce yourself (or both members, if working in a pair) and state the purpose of the project.
- **Overview:** Provide a quick overview of what you'll be covering in the video:
  - The selected dataset and object detection algorithm (HOG-SVM, YOLO, or SSD).
  - The main steps in the project: data preparation, model implementation, training, evaluation, and submission.

2. Data Preparation (2-3 minutes)

- **Dataset Description:** Show the dataset you are using. Explain why it was chosen and what kind of images or objects it contains.
- **Preprocessing Steps:** Walk through the data preprocessing steps:
  - How images were resized or normalized.
  - How bounding boxes or labels were applied to the images (if applicable).
- **Screen Recording:** Use a screen recording tool to show your code or scripts for data preprocessing. Briefly explain key parts of the code.

3. Model Implementation (3-4 minutes)

- **Selecting the Algorithm:** State the algorithm you selected (HOG-SVM, YOLO, SSD) and why you chose it.
- **Building the Model:** Walk through the implementation process:
  - Highlight the code sections where the model is constructed and configured.
  - Discuss any specific libraries or frameworks used (e.g., TensorFlow, OpenCV).
- **Explanation:** Explain important aspects of the code, such as:
  - Feature extraction (for HOG-SVM).
  - Neural network architecture (for YOLO or SSD).
- **Tips:** Use comments within your code to make it easier to explain during the recording.

4. Training the Model (2-3 minutes)

- **Dataset Splitting:** Describe how you split the dataset into training and testing sets.
- **Training Process:** Show the code for training the model, explaining the key parameters you tuned (e.g., learning rate, batch size, epochs).
- **Output:** Display part of the training output (e.g., accuracy, loss) and describe how the model is performing during the training process.

5. Testing and Evaluation (2-3 minutes)

- **Testing:** Show how you tested the model using the test dataset.
- **Evaluation Metrics:** Discuss the evaluation metrics used (accuracy, precision, recall, speed) and explain what they indicate about your model's performance.
- **Visual Results:** Display images where the model successfully detected objects, using bounding boxes and labels.
- **Comparison:** If applicable, compare the results of different algorithms (e.g., HOG-SVM vs. YOLO).

6. Discussion of Challenges (1-2 minutes)

- **Challenges:** Discuss any challenges you faced during the project (e.g., data preprocessing issues, model tuning difficulties) and how you overcame them.
- **Learning:** Briefly mention any key learnings or insights you gained while working on the project.

7. Conclusion and Next Steps (1-2 minutes)

- **Summarize:** Recap the main parts of the project: dataset selection, model implementation, training, and evaluation.
- **Next Steps:** Suggest potential improvements or future work (e.g., using a larger dataset, trying different algorithms).
- **Thank You:** End the video with a closing remark, thanking the viewer for their time.

Tips for Video Recording:

- **Use Screen Recording Software:** Tools like OBS Studio, Camtasia, or Zoom can be used to record your screen. Ensure you also record audio to explain each step.
- **Be Clear and Concise:** Keep explanations straightforward, especially when walking through code.
- **Highlight Important Sections:** Use your cursor or on-screen annotations to highlight parts of the code or output you are discussing.
- **Practice:** Run through what you will say a couple of times before recording to ensure a smooth and coherent presentation.

- **Edit the Video:** Trim unnecessary parts to keep the video within the 10-15 minute range and ensure a focused presentation.

By following these guidelines, your video documentation will provide a comprehensive view of the project from start to finish, showcasing your understanding of object detection using machine learning.

Rubric for Mid-term Project: Implementing Object Detection on a Dataset

Criteria	**Excellent (90-100%)**	**Good (75-89%)**	**Satisfactory (60-74%)**	**Needs Improvement (0-59%)**
Dataset and Algorithm Choice	Well-justified selection of dataset and algorithm, demonstrating a deep understanding of their suitability. Clear explanation of the dataset's features and algorithm's advantages.	Appropriate selection with some reasoning provided, but lacks detailed explanation of suitability.	Basic selection with minimal reasoning. Explanation of suitability is brief.	Poor or missing justification. No clear explanation of dataset or algorithm choice.
Implementation	Correct and efficient implementation, with well-organized, well-commented code. Utilizes advanced techniques and demonstrates strong coding practices.	Mostly correct implementation, with minor issues in efficiency or structure. Code has some comments and is fairly organized.	Basic implementation with functional code, but lacks optimization, comments, and clear structure.	Incorrect or missing implementation. Poorly structured code with no comments.
Model Training	Thorough training with detailed parameter tuning and optimization. Shows critical thinking in improving model performance. Includes discussion on why specific hyperparameters were chosen.	Adequate training and parameter tuning, but lacks depth in explaining choices. Some optimization present.	Basic training with minimal parameter tuning. Lacks depth in optimization and explanation.	Poor training with little to no parameter tuning. Results are unclear or inaccurate.
Evaluation	Comprehensive evaluation using all specified metrics (accuracy, precision, recall, speed). In-depth analysis of performance, with insightful conclusions and comparison with other methods.	Evaluation mostly correct, but misses some metrics or lacks depth in analysis. Some comparison included.	Basic evaluation present but lacks depth or misses key metrics. Minimal comparison provided.	Inadequate or missing evaluation with incorrect or unclear analysis. No comparison.
Report	Detailed and well-organized report with clear explanations, visualizations, and critical reflections. Covers all aspects of the project, including challenges and next steps.	Clear report but lacks detail in some areas. Covers most aspects but with minor structural issues.	Basic report present, with limited explanations. Some key information is missing.	Poor or missing report. Lacks clarity, detail, and critical analysis.
Video Documentation	Engaging and clear video (10-15 minutes) that provides a comprehensive overview. Includes a detailed walkthrough of the implementation, training process, results, and challenges. Excellent use of screen recordings and annotations.	Video is mostly clear and informative (7-10 minutes), but lacks depth in certain areas. Walkthrough is present but may skip some details.	Basic video present (5-7 minutes) with limited information and unclear explanations. Minimal use of screen recording.	Poor or missing video documentation. Video is under 5 minutes, lacks key elements, or is difficult to follow.
Code Quality	Code is efficient, follows best practices, well-documented, and easy to understand. Makes use of functions, modular code, and appropriate libraries.	Code works and follows some best practices. Contains comments but could be more detailed. Minor areas for optimization.	Functional code, but lacks structure and detailed comments. Limited use of functions or modular code.	Code is incorrect, poorly structured, lacks comments, and contains inefficient practices.

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (60-74%)	Needs Improvement (0-59%)
Visualization of Results	Clear, well-labeled images showing accurate object detection. Provides multiple examples for robust evaluation. Visuals are neatly organized and easy to interpret.	Visuals are mostly clear, but lack some labels or details. Provides sufficient examples for evaluation.	Basic visuals present but lack labeling and clarity. Minimal examples shown.	Poor or missing visualization. Images are unclear, lack labels, or do not demonstrate proper results.
Challenges and Solutions	Thoughtful discussion of challenges faced, with detailed explanations of how they were addressed. Reflects critical thinking and problem-solving skills.	Discussion is present, but lacks depth or misses some key challenges. Provides basic solutions.	Basic mention of challenges, with minimal reflection on solutions or how they were addressed.	Lacks discussion of challenges or how they were handled. No reflection on the problem-solving process.
File Organization and Submission	All files (code, images, documentation, video) are correctly named and organized according to submission guidelines. Proper use of folders and subfolders.	Mostly follows naming and organization requirements, with minor errors.	Basic organization, but does not fully adhere to the specified format. Some files are misplaced or incorrectly named.	Poor organization, incorrect file names, missing files, or does not follow the submission guidelines.

This mid-term project allows students to apply their knowledge of object detection in machine learning, gain hands-on experience with different algorithms, and critically assess their performance.